Remarks/Arguments

Applicant has received and carefully reviewed the Final Office Action of the Examiner mailed April 7, 2004. Claims 1-27 remain pending. Reexamination and reconsideration are respectfully requested.

In paragraph 44 of the Final Office Action, the Examiner states that Applicants previous remarks were not persuasive. In paragraph 46 of the Office Action, the Examiner states that Nakayama has taught that "it is necessary to set the external conditions of the binary floating point data, which is obtained as a result of the arithmetic operation, in a status register which can be checked by software" (citing Nakayama, column 1, lines 9-13). The Examiner also states that a conditional branch, also known as a conditional jump in the art, is a software instruction which checks the status register before transferring program control to another instruction. The Examiner further states that the conditional field set in Nakayama, which indicates whether a certain condition is satisfied or not, is used by software, specifically a software conditional branch instruction.

As the Examiner knows, "[a] claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). [see also, MPEP § 1231]. To establish inherency, the extrinsic evidence must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill. Inherency, however, may not be established by probabilities or

possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient. [see MPEP § 1212].

The Examiner appears to be arguing that the statement in Nakayama that "it is necessary to set the external conditions of the binary floating point data, which is obtained as a result of the arithmetic operation, in a status register which can be checked by software (citing Nakayama, column 1, lines 9-13), necessarily discloses "generating at least one status bit based on the digital value to be stored, the at least one status bit indicating if a predetermined condition of a conditional jump instruction is satisfied, as recited in claim 1. More specifically, the Examiner appears to be arguing that because Nakayama states that the status register is checked by "software", and that a conditional branch instruction is a software instruction, that the status register must be checked by a conditional branch instruction. This reasoning, however, appears to be improper.

As noted previously, Nakayama never mentions a conditional "branch" or "jump" instruction, and as such, the Examiner must be relying on the doctrine of inherency to make the rejection. However, as the Examiner is undoubtedly aware, there are numerous types of "software" instructions, only some of which are conditional branch or jump type instructions. As such, Nakayama clearly does not necessarily check the status register with a conditional branch instruction, as suggested by the Examiner. As noted above, inherency may not be established by probabilities or possibilities. Further, the mere fact that a certain thing may result from a given set of circumstances is not sufficient. [see MPEP § 1212].

In addition to the foregoing, nowhere do Nakayama et al. appear to disclose or suggest generating at least one status bit that indicates if a predetermined condition of a

conditional jump instruction is satisfied, as recited in claim 1. As noted previously, Nakayama never mentions a conditional "branch" or "jump" instruction, and therefore cannot disclose or suggest "generating at least one status bit that indicates if a predetermined condition of a conditional jump instruction is satisfied. As noted in the present specification:

> The present invention overcomes many of the disadvantages of the prior art by providing a branch prediction method and system that can accurately predict a branch condition early in the instruction pipeline, and preferably before the instruction reaches the arithmetic stage of the instruction processor. Thus, when the present branch prediction method is applied, the correct target instruction can be fetched early, thereby avoiding many of the inefficiencies associated with branch miss predictions.
>
> In a preferred embodiment, the present invention is used in conjunction with a pipelined instruction processor that executes instructions including conditional jump instructions. As is known, conditional jump instructions typically read a digital value from memory to determine if the condition of the conditional jump instruction is satisfied. To accurately determine if the conditions are satisfied, the memory preferably stores one or more pre-calculated status bits along with the digital value that is read by the conditional jump instruction to determine if the condition is satisfied. Each status bit corresponds to a particular type of jump instruction, and indicates whether the particular jump condition is satisfied. For example, one status bit may correspond to a jump zero condition that jumps to a target address if the digital value read from the memory equals zero. In this case, the status bit may indicate if the corresponding digital value is equal to zero.
>
> By including such a status bit, the condition of the conditional jump instruction may be immediately determined, without waiting for the instruction to be processed by the arithmetic unit or the like in a subsequent pipeline stage. Other illustrative conditional jump instructions include, for example, jump positive, jump negative, and jump low bit instructions.

(Specification, page 3, line 11 though page 4, line 10). In contrast the foregoing, the "condition" of Nakayama appears to identify attributes of binary floating point data of source and destination operands so that binary floating point arithmetic operations are

performed properly. In view of the foregoing, Applicant does not see how it can readily be argued that Nakayama discloses or suggests generating at least one status bit that indicates if a predetermined condition of a conditional jump instruction is satisfied, as recited in claim 1.

The undersigned participated in an interview with the Examiner on June 7, 2004. The undersigned would like to thank the Examiner for the courtesies extended during the interview. During the interview, the Examiner acknowledged that the references used to reject the present claims do not function in a similar way to the present invention, but that when the language of the claims is interpreted very broadly, the claims cannot be considered patentable over the cited references. For example, during the interview and with respect to claim 1, the Examiner stated that the word "indicating" is very broad, meaning merely "to point out or point to: to be a sign, symptom, or index of". Thus, the Examiner reasoned that, a sign bit, for example, could provide an "indication" of whether a predetermined condition of a conditional jump instruction is satisfied.

While Applicants respectfully disagree with the Examiner's conclusions for the reasons set forth above as well as other reasons, claim 1 has been amended to recite:

> 1. (Currently Amended) A method for storing a digital value to memory in a pipelined instruction processor, wherein the digital value is read from memory in response to a conditional jump instruction to determine if the condition of the conditional jump instruction is satisfied, the method comprising:
>
> generating at least one status bit based on the digital value to be stored, the at least one status bit relating to a particular condition of a conditional jump instruction and specifying ~~indicating~~ if the particular ~~a predetermined~~ condition of [[a]] the conditional jump instruction is satisfied or not; and
>
> storing the digital value and the at least one status bit to memory.

As can be seen, claim 1 has been amended to recite that the at least one status bit relates to a particular condition of a conditional jump instruction, and specifies if the particular condition of the conditional jump instruction is satisfied or not (Emphasis Added). The word "specify" is defined as "to name or state explicitly or in detail" (see, Merriam-Webster's Dictionary). Nothing in the cited prior art suggests generating at least one status bit that relates to a particular condition of a conditional jump instruction, and further specifies if the particular condition of the conditional jump instruction is satisfied or not. In view thereof, claim 1 is believed to be clearly patentable over the cited prior art. For similar and other reasons, dependent claims 2-9 are also believed to be clearly patentable over the cited prior art.

In addition, claim 2 has been amended to recite that the conditional jump instruction reads the digital value and the at least one status bit from memory to determine if the condition of the conditional jump instruction is satisfied without having to submit the condition of the conditional jump instruction to an arithmetic logic stage of the pipelines instruction processor. During the interview, the Examiner acknowledged that the cited prior art fails to suggest using a status bit to determine if a condition of a conditional jump instruction is satisfied without having to submit the condition of the conditional jump instruction to an arithmetic logic stage of the pipelines instruction processor. For these additional reasons, claim 2 is believed to be clearly patentable over the cited prior art.

For similar reasons to those given above, as well as other reasons, claims 10-18, as amended, are also believed to be clearly patentable over the cited art.

In paragraph 23 of the Final Office Action, the Examiner rejected claims 19-22 and 24-27 under 35 U.S.C. § 103(a) as being unpatentable over Watson et al. (U.S. Patent No. 3,573,854) in view of Nakayama et al. In paragraph 25 of the Final Office Action, the Examiner states that Watson et al. do not teach or suggest: (a) one or more of the conditional jump instructions reading a digital value from memory to determine if the condition of the conditional jump instruction is satisfied; (b) storing a value that includes a digital value and at least one jump status bit; and (c) wherein the current instruction includes an address and a corresponding jump field, the address identifies one of the addressable registers and the corresponding jump field identifies a jump status bit of the at least one jump bits within the identified addressable register.

While Applicants agree with the Examiner that Watson et al. do not teach or suggest these elements, Applicants also believe that Watson et al. do not teach or suggest many other elements of claim 19, including the jump look ahead controller, the tracking logic and the conflict detection logic as recited.

The Examiner then states that Nakayama et al. suggest: (a) one or more of the conditional jump instructions reading a digital value from memory to determine if the condition of the conditional jump instruction is satisfied; (b) storing a value that includes a digital value and at least one jump status bit; and (c) wherein the current instruction includes an address and a corresponding jump field, the address identifies one of the addressable registers and the corresponding jump field identifies a jump status bit of the at least one jump bits within the identified addressable register (citing Nakayama et al.: Abstract; column 1, lines 9-13 and 49-68; column 2, lines 20-50, Figure 1A; Figure 1B; Figure 5A; and Figure 5B).

Claim 19 recites:

> 19. (Previously Presented) In a pipelined instruction processor that executes instructions including conditional jump instructions, one or more of the conditional jump instructions reading a digital value from memory to determine if the condition of the conditional jump instruction is satisfied, the improvement comprising:
>
> a plurality of addressable registers, each of the addressable registers storing a value that includes a digital value and at least one jump status bit;
>
> logic to access a current instruction, wherein the current instruction includes an address and a corresponding jump field, the address identifies one of the addressable registers and the corresponding jump field identifies a jump status bit of the at least one jump status bits within the identified addressable register;
>
> a jump look-ahead controller for generating a jump look-ahead signal using the address that identifies one of the addressable registers and the jump field that identifies a jump status bit within the identified addressable register, the jump look-ahead signal is a function of the identified jump status bit;
>
> tracking logic for tracking the addresses of a predetermined number of previous instructions in the pipelined instruction processor and comparing the addresses of each previous instruction to the address of the current instruction to generate a series of jump disable signals; and
>
> conflict detection logic for generating a jump early signal using the jump look-ahead signal and the series of jump disable signals, the jump early signal initiates the conditional jump depending on the values of the jump disable signals.

(Emphasis Added). As previously noted, Nakayama et al. do not appear to relate in any way to conditional jump instructions, and in particular, one or more conditional jump instructions that read a digital value from memory to determine if the condition of the conditional jump instruction is satisfied. In addition, Nakayama et al. do not appear to relate in any way to storing a value that includes a digital value and at least one jump status bit (Emphasis Added). Finally, Nakayama et al. do not appear to relate in any way to a system whereby the current instruction includes an address and a corresponding jump field, where the address identifies one of the addressable registers and the corresponding

jump field identifies a jump status bit of the at least one jump bits within the identified addressable register (Emphasis Added). As noted above, the terms "branch" and "jump" are never even mentioned in Nakayama et al.

In response to the above, and in paragraph 47 of the Final Office Action, the Examiner states that Watson has taught in column 7, line 7 to column 8, line 13 how conditional branches are handled, including accessing a jump status bit, also known as the condition. The Examiner states that the cited lines allude to accessing the condition via an address and the jump field. The Examiner further states that it is inherent that the condition would be accessed using an address and the jump field, since that is the only way the system would know where to look in memory and what condition to specifically check for.

After careful review, Applicants must respectfully disagree. Nakayama et al. do not appear to disclose or suggest many of the elements of claim 19 including, for example, providing a current instruction that includes an address and a corresponding jump field, where the address identifies one of the addressable registers and the corresponding jump field identifies a jump status bit of the at least one jump bits within the identified addressable register (Emphasis Added). As discussed during the interview, Nakayama et al. may generate an external condition to identify attributes of the binary floating point data of source and destination operands so that binary floating point arithmetic operations are performed properly. However, this is completely different than, for example, having a corresponding jump field identify a jump status bit of the at least one jump bits within the identified addressable register (Emphasis Added) (see also, for example, Figure 9 of the present specification). That is, the condition of Nakayama et al.

does not identify a jump status bit of at least one jump bits within an identified addressable register, as recited in claim 19. Nor would it be inherent for the condition of Nakayama et al. to be accessed using an address AND the jump field. In Nakayama et al., there does not appear to be a jump field, only a "condition". In fact, it appears that the "condition" of Nakayama et al. is generated on the fly and not read up from memory, as the Examiner appears to be suggesting (see, for example, Figure 7 of Nakayama et al. and the corresponding description at column 8, lines 22-39).

For these and other reasons, claim 19 is believed to be clearly patentable over Watson et al. in view of Nakayama et al. *If the Examiner elects to maintain this rejection, Applicants respectfully request that the Examiner specifically point out where each and every element of claim 19 is disclosed or suggested in the cited prior art.* For similar and other reasons, dependent claims 20-26 are also believed to be clearly patentable over Watson et al. in view of Nakayama et al.

Turning now to claim 27. For the reasons set forth in the Amendment filed on January 20, 2004, as well as other reasons, claim 27 is believed to be clearly patentable over the cited prior art.

In paragraph 36 of the Final Office Action, the Examiner states that Watson et al. do not teach or suggest: (a) storing a digital value and one or more jump status bits that are based on the digital value in each of a plurality of address locations in an addressable memory; and (b) accessing a current instruction, the current instruction having an address and a jump field, the address identifies a selected address location of the addressable memory, and the jump field identifies a selected jump status bit of the selected address location. However, the Examiner states that Nakayama et al. suggest these elements.

After careful review, Applicants must respectfully disagree for many reasons. For example, Nakayama et al. do not appear to disclose or suggest providing a current instruction that has an address and a jump field, wherein the address identifies a selected address location of the addressable memory, and the jump field identifies a selected jump status bit of the selected address location (Emphasis Added). The Examiner does not appear to address this and other language of claim 27 in the Final Office Action. *If the Examiner elects to maintain this rejection, Applicants respectfully request that the Examiner specifically point out where Nakayama et al. discloses or suggests all of the limitations of claim 27 including, for example, a current instruction that includes an address and a jump field, wherein the jump field identifies a selected jump status bit of the selected address location, as recited in claim 27.* For these and other reasons, claim 27 is believed to be clearly patentable over the cited prior art.

In view of the foregoing, it is believed that all pending claims 1-27 are now in condition for allowance. Issuance of a notice of allowance in due course is respectfully

requested. If a telephone conference would be of assistance, please contact the undersigned attorney at 612-359-9348.

Respectfully submitted,

Lawrence R. Fontaine et al

By their attorney,

Dated: July 7, 2004

Brian N. Tufte, Reg. No. 38,638
CROMPTON, SEAGER & TUFTE, LLC
1221 Nicollet Avenue, Suite 800
Minneapolis, MN 55403-2402
Telephone: (612) 677-9050
Facsimile: (612) 359-9349